Technology Magic of Keynote

Session 202

Technology Magic of Keynote

Travis W. Brown
Graphics & Imaging Evangelist
Worldwide Developer Relations

Why Talk About an Application

- Shows Technology in Action
- Whole is greater than sum of the individual parts
- Articulates the possibilities

Why Talk About Keynote?

- Keynote is platform compelling
  - Simple to use, yet incredibly powerful
  - Fully leverages many important Mac OS X technologies
    - Cocoa, Quartz 2D, QuickTime, OpenGL, XML
    - These technologies can be used in your apps as well
  - Learn from a real world development cycle

A Quick Tour of Keynote

Brad Vaughan
Keynote Engineering Manager
Productivity Engineering

What You'll Learn

- Keynote—powered by Cocoa
- Cocoa text: tips and tricks
- Quartz and image management
- OpenGL for 2D and 3D effects
- QuickTime media control
- Open file formats with XML

Application Frameworks

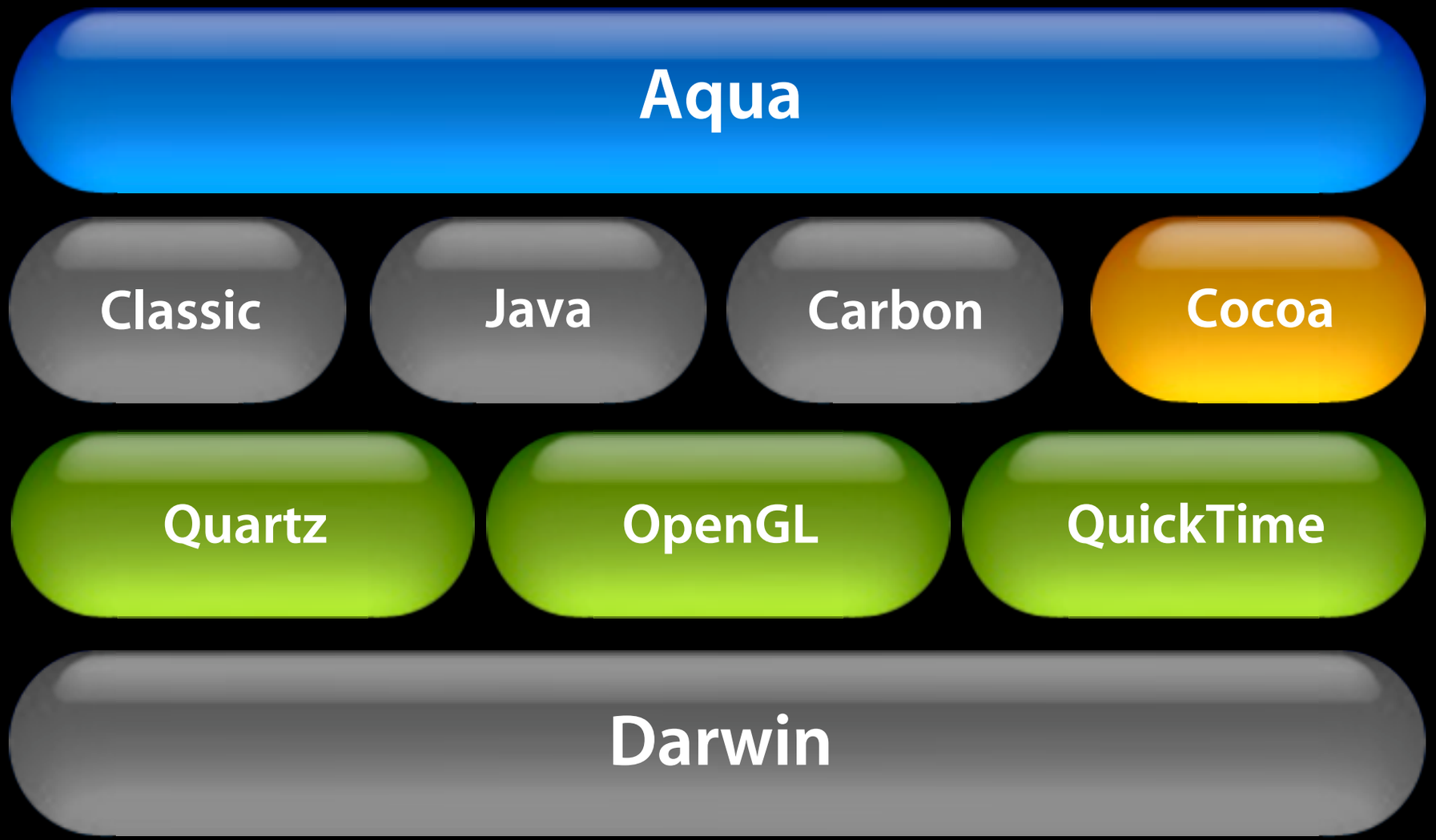
Considerations

- First-class integration
- Powerful components
- Flexible
- Extensible
- Performance
- Ease of development

Cocoa

Cocoa Advantages

Integration



Cocoa Advantages

Features

- Standard, flexible UI controls
- NSImage—QuickTime compatibility
- Complex views: NSOpenGLView, NSMovieView
- NSDocument
- Localization
- Runtime support

Cocoa Advantages

Ease of development

- Objective C
  - Powerful
  - Simple
- ProjectBuilder
  - Full-feature IDE
  - SCM integration

Cocoa

Summary

- First-class integration
- Powerful components
- Best of breed tools

Mac OS X Text

Overview

Cocoa Undo, keybindings

ATSUI Unicode, typographic features

Quartz Antialiasing

Demo

Cocoa Text

Advanced features

Keynote → Keynote

日本語

Zapfino → *Zapfino*

Cocoa Text

Keynote challenge

White Theme

Cocoa Text

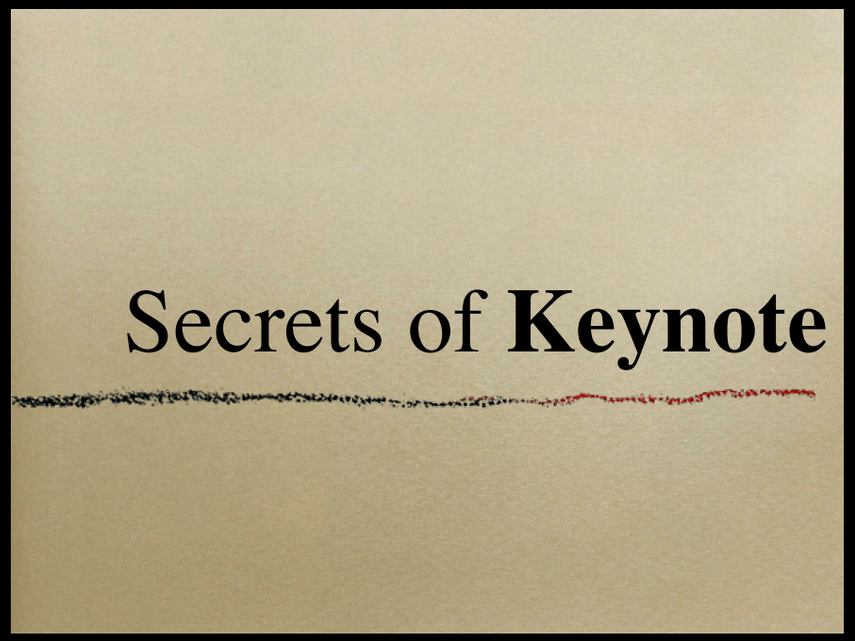
Keynote challenge

White Theme

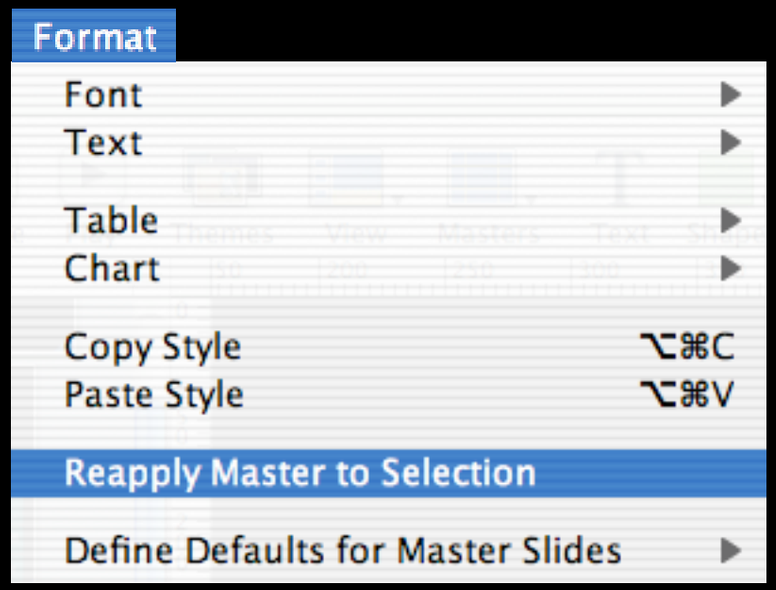
Crayon Theme

Cocoa Text

Keynote challenge

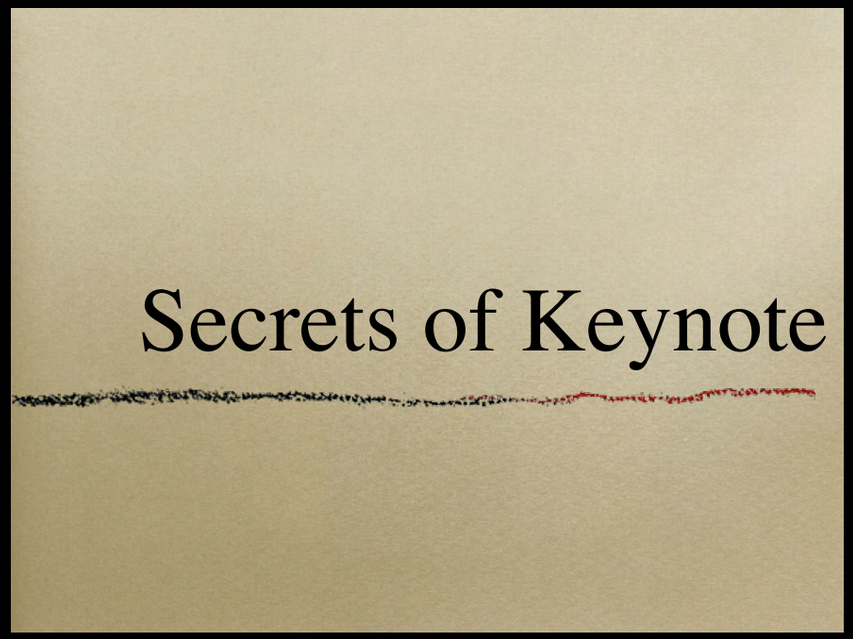


Crayon Theme

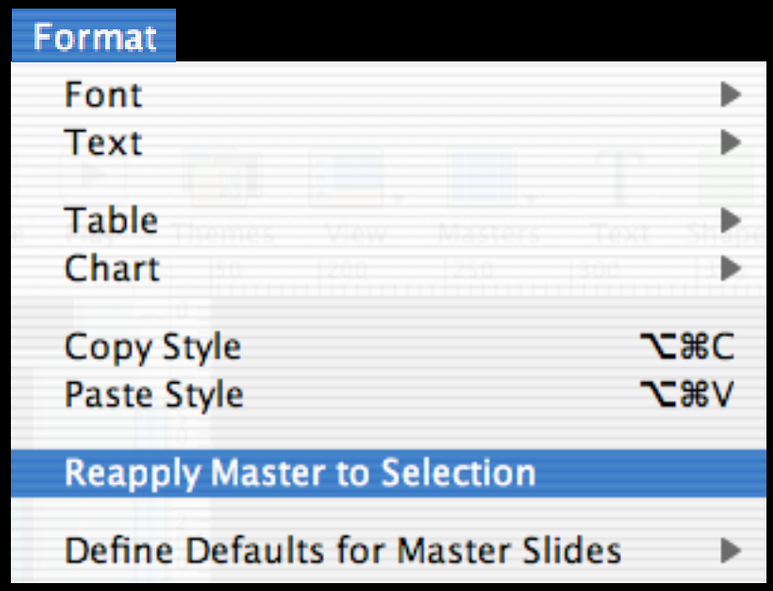


Cocoa Text

Keynote challenge

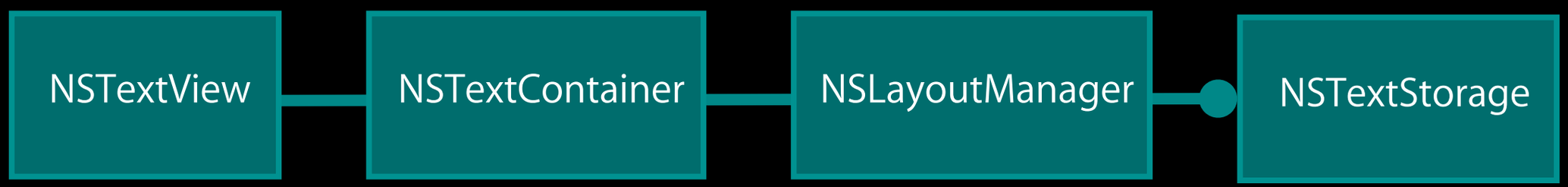


Crayon Theme



Cocoa Text

Handling Cascading Styles

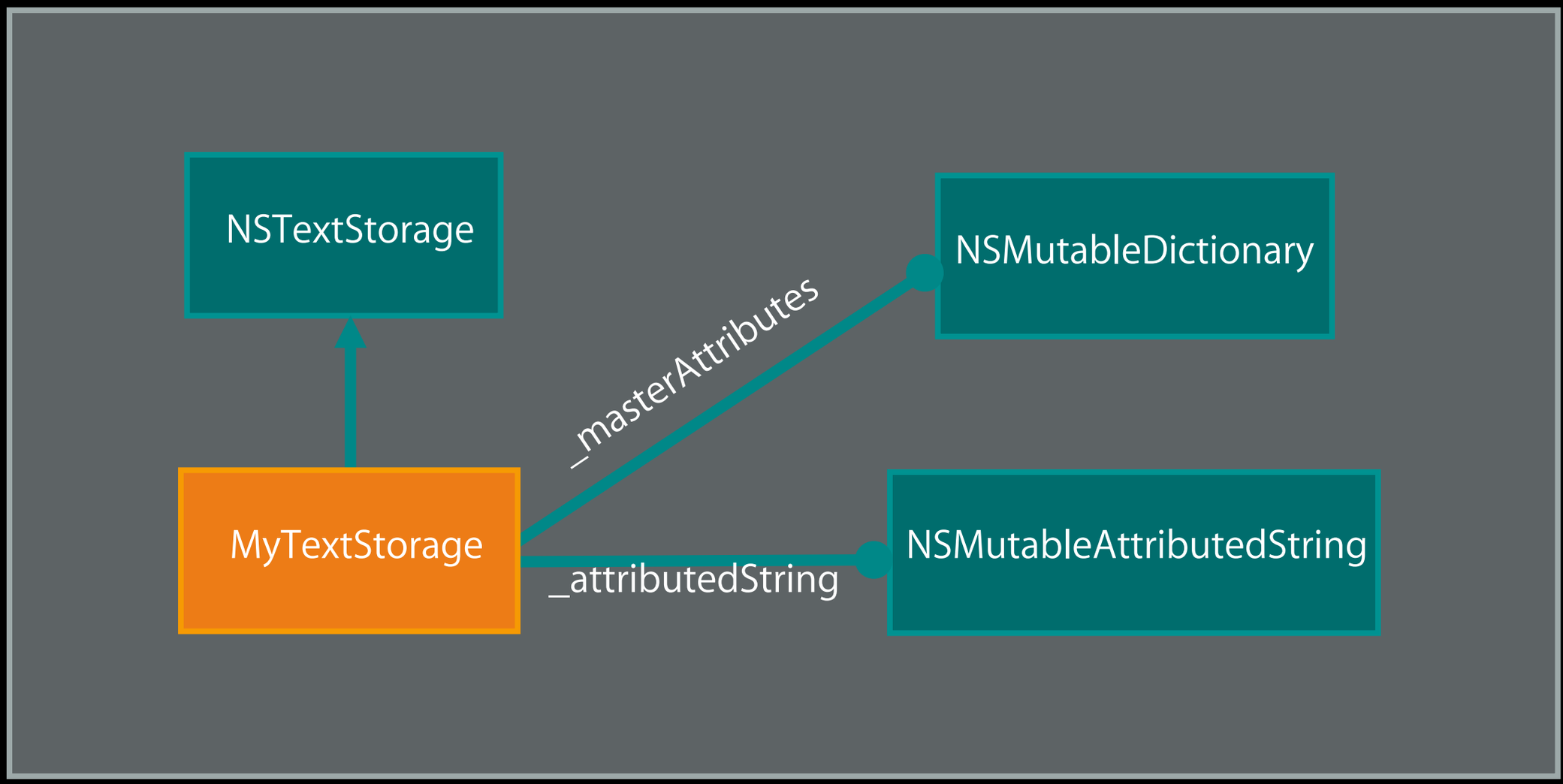



```
// Modifies a run of text in an attributed string
- (void)setAttributes:(NSDictionary *)attrs
                range:(NSRange)range;
```

Cocoa Text

Handling Cascading Styles

- Solution: NSTextStorage subclass



Cocoa Text

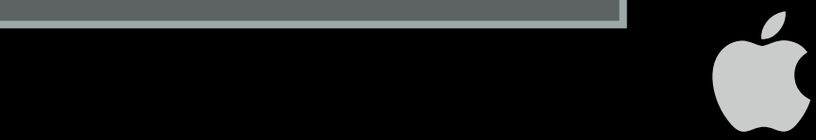
Handling Cascading Styles

```
// Required NSTextStorage overrides are forwarded to
// _attributedString (our data storage)

- (NSString *)string;

- (NSDictionary *)attributesAtIndex:(unsigned)index
      effectiveRange:(NSRangePointer)aRange;

- (void)replaceCharactersInRange:(NSRange)aRange
      withString:(NSString *)aString;

- (void)setAttributes:(NSDictionary *)attributes
   range:(NSRange)aRange;
```

Cocoa Text

Handling Cascading Styles

```
- (id)initWithAttributedString:(NSAttributedString*)str
    masterAttributes:(NSDictionary *)dict;

// Used for menu validation
- (BOOL)doesOverrideMasterAttributes;

// “Revert overrides”...
- (void)reapplyMasterAttributes;

// Sets master attributes. Cascades change to
// _attributedString by walking all ranges of text,
// replacing old master styles.
// See attributesAtIndex:longestEffectiveRange
- (void)setMasterAttributes:(NSDictionary *)dict;
```



Cocoa Text

Summary

- Unparalleled feature set out of the box
- Minimal coding
- UI support

Quartz 2D

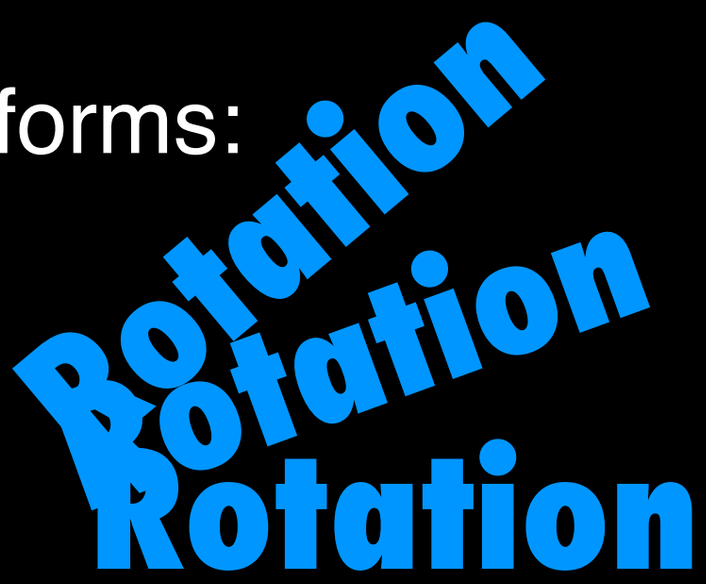
Overview

Paths:

Images:

PDF:

Transforms:

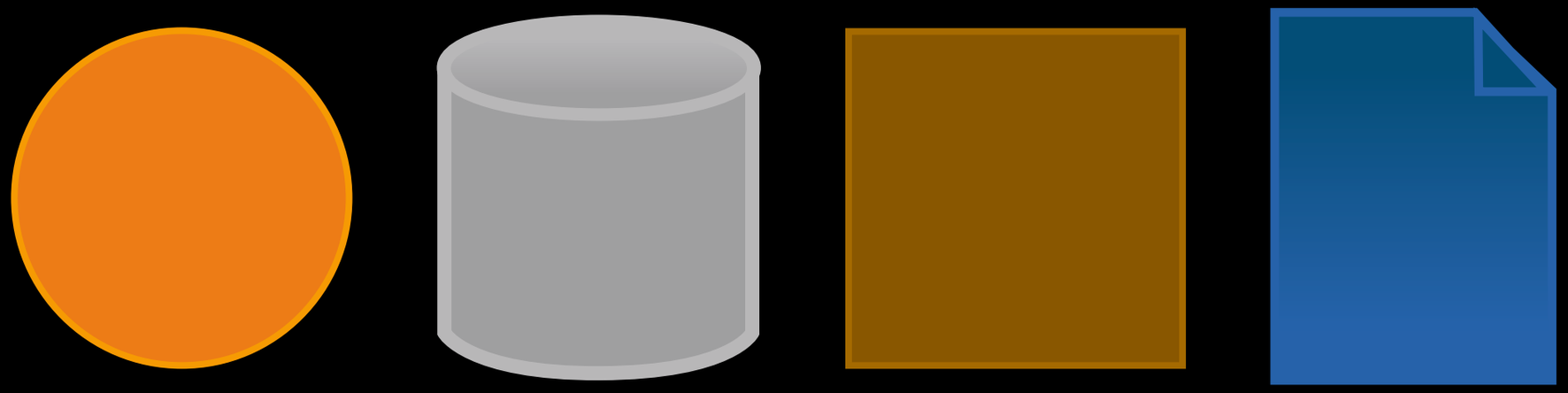


Transparency:

Demo

Quartz

Features

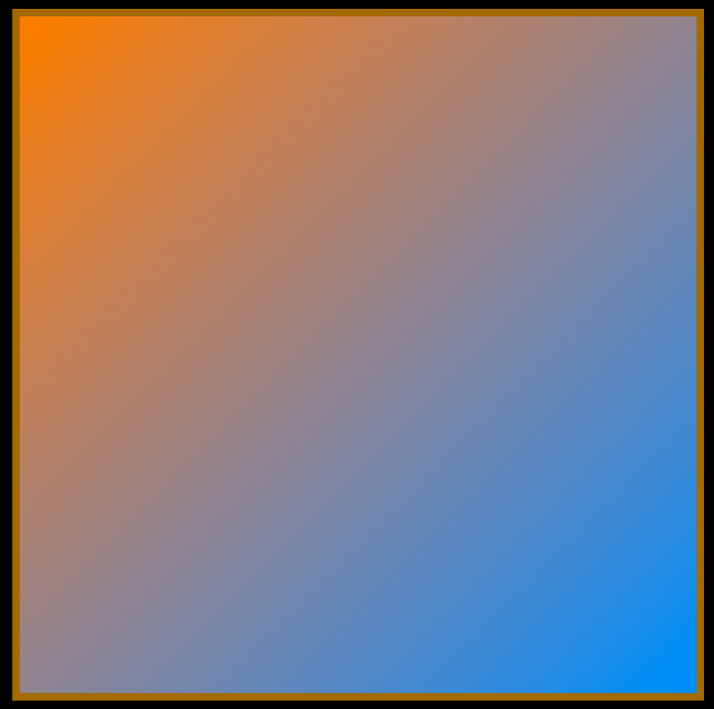


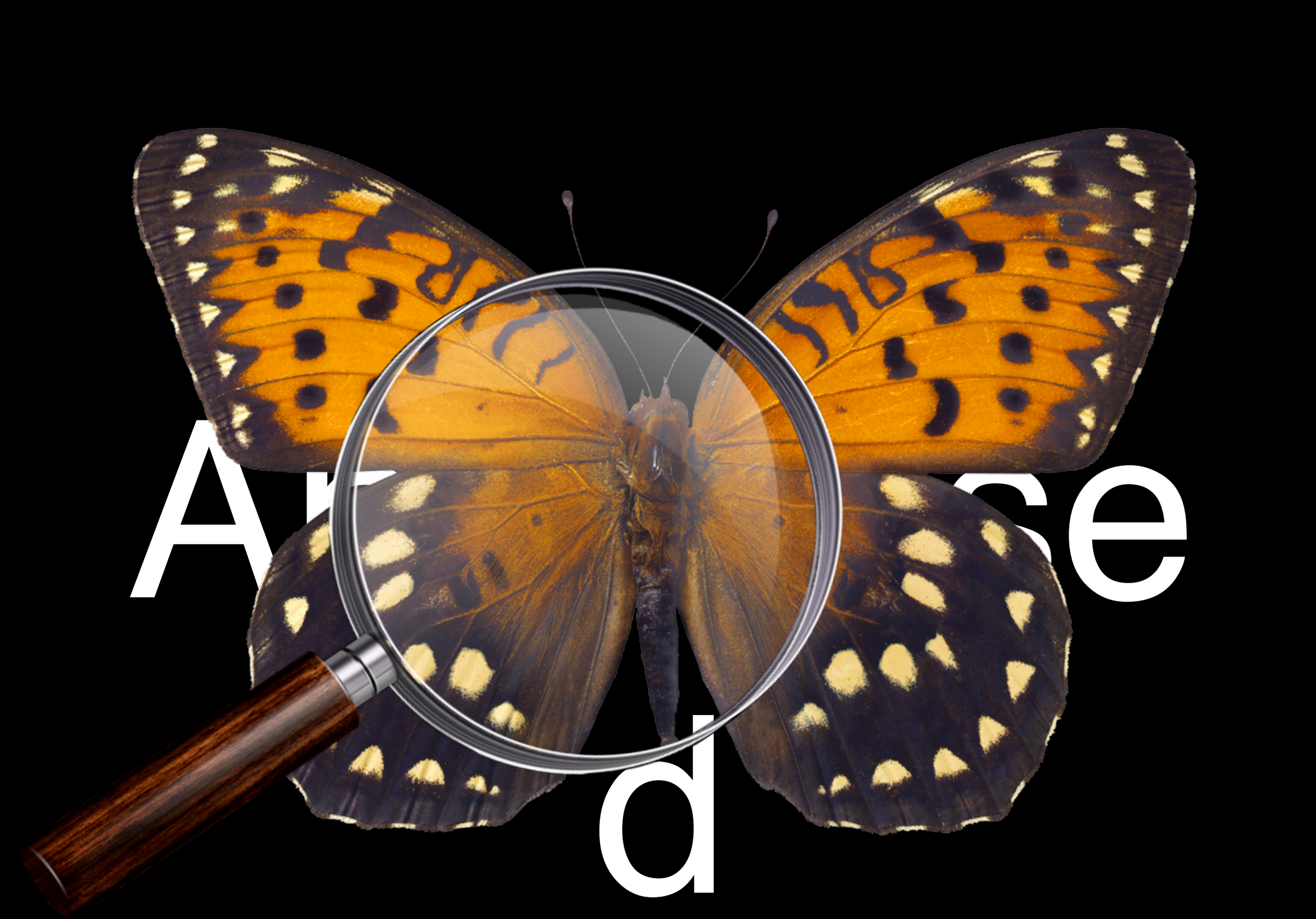
Quartz

Features

Quartz

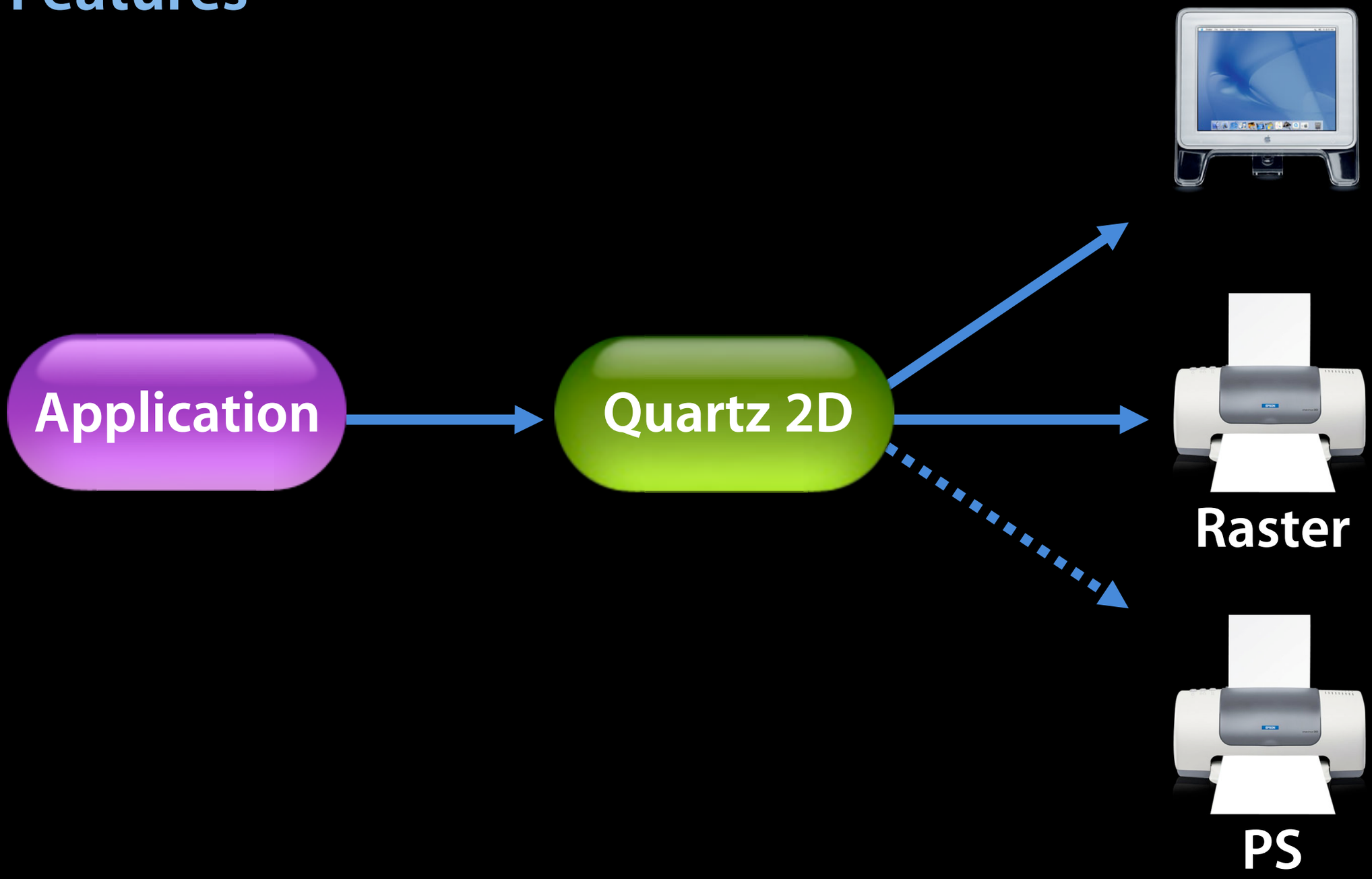
Features




A
se
d

Quartz 2D

Features



Quartz

Features

Quartz

Keynote challenge

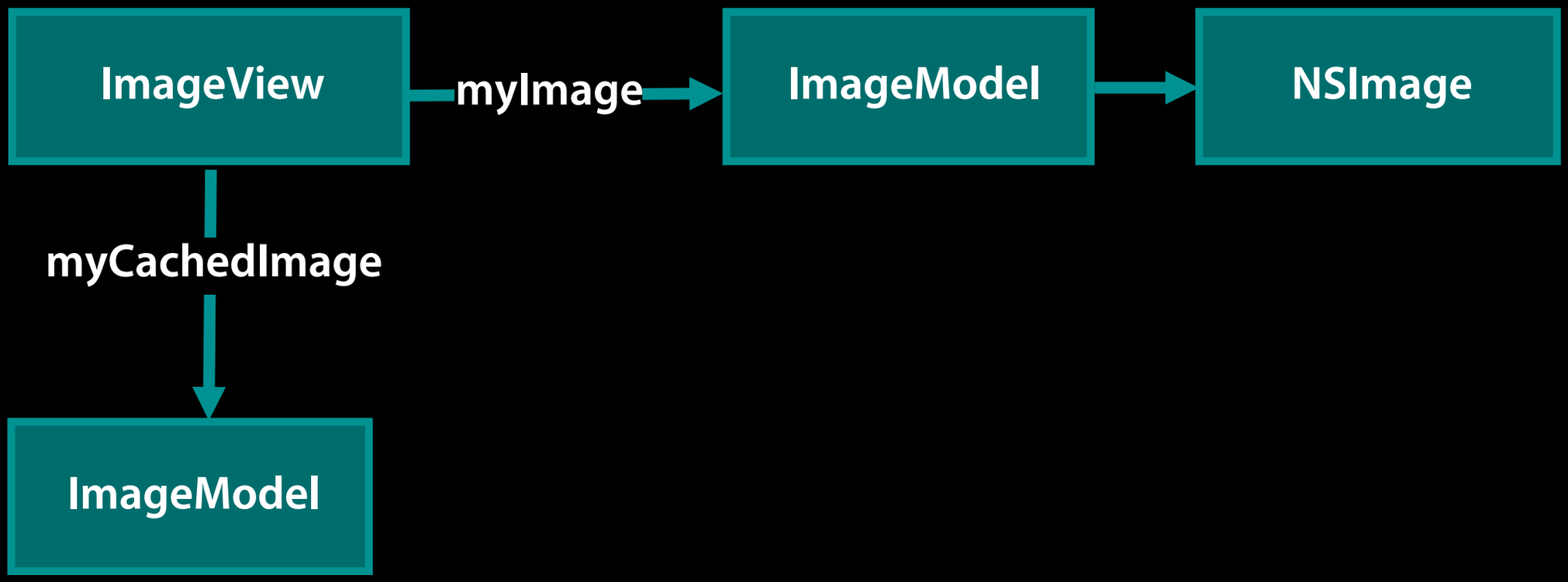
- Key feature—Professional quality image manipulation
  - Fast manipulation of multi-megabyte images
  - Quickly switching between slides with scaled images

High-quality resize

```
// Cocoa
context = [NSGraphicsContext currentContext];
[context setImageInterpolation:NSImageInterpolationHigh];

// CoreGraphics
CGContextSetInterpolationQuality([context graphicsPort], kCGInterpolationHigh);
```

Quartz

Image caching



Quartz
Image caching

```
@implementation SlideCanvas

- (void)mouseDown:(BOOL)flag {
    // ...
    // If the selected object is an image, tell it to cache on resize.
    [selectedImageView setDynamicallyResizing:YES];
    // renders all affected objects...
}
```

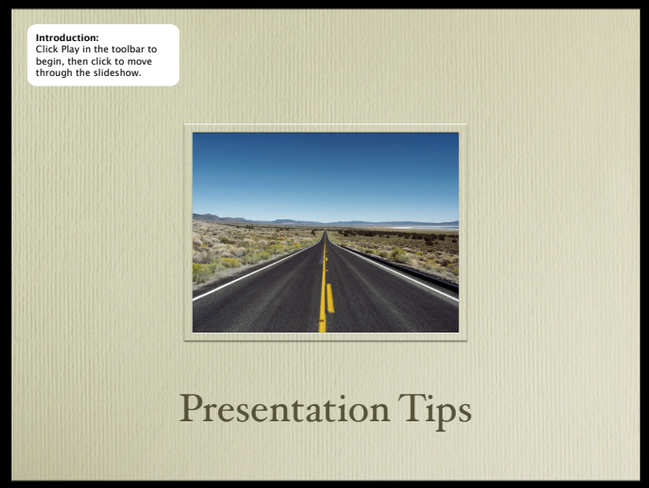
Quartz
Image caching

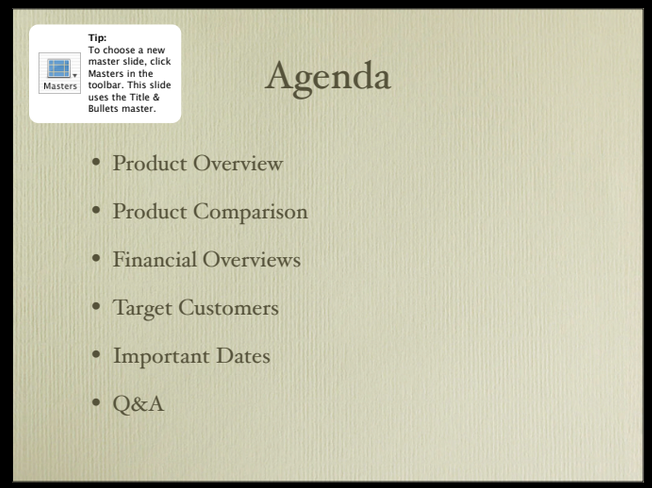
```
@implementation ImageView

- (void)setDynamicallyResizing:(BOOL)flag {
   if (flag && NSWidth([self bounds]) / [imageModel width] < 0.9) {
      [self setCachedImage:[imageModel copyWithScaleX:xFactor
                                               scaleY:yFactor];
   } else {
      [self setCachedImage:nil];
   }
}

- (void)render {
   if (nil != [self cachedImage]) {
      // the graphics context is transformed and restored by the sender
      [myCachedImage drawInRect:[self bounds]];
   } else {
      [myImage drawInRect:[self bounds]];
   } // ...
```

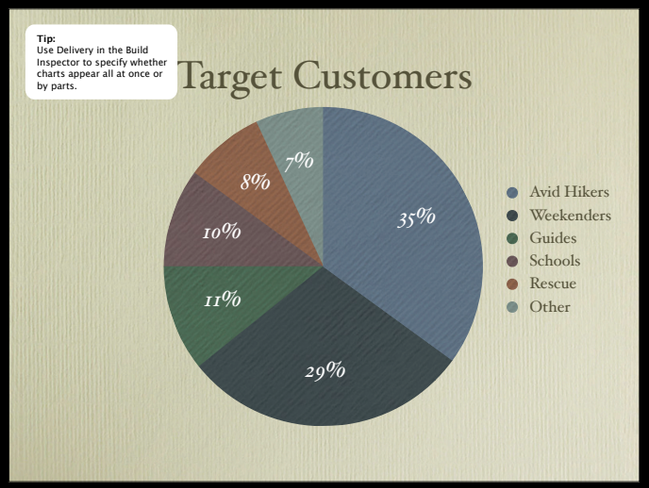
Quartz

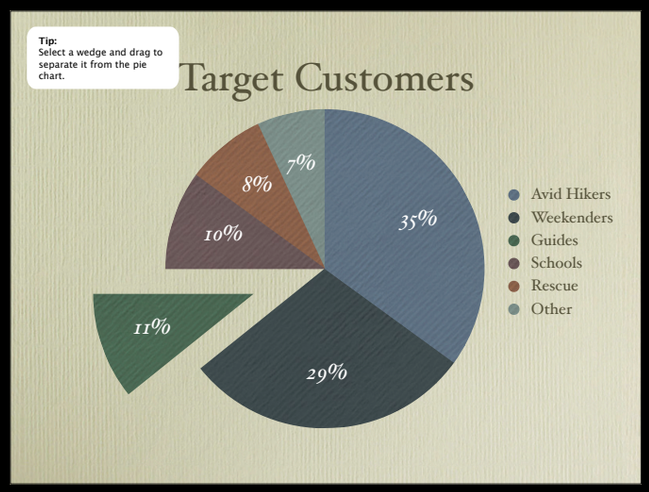
Keynote challenge

- Efficient production of multi-page PDF with many recurring, high-quality images







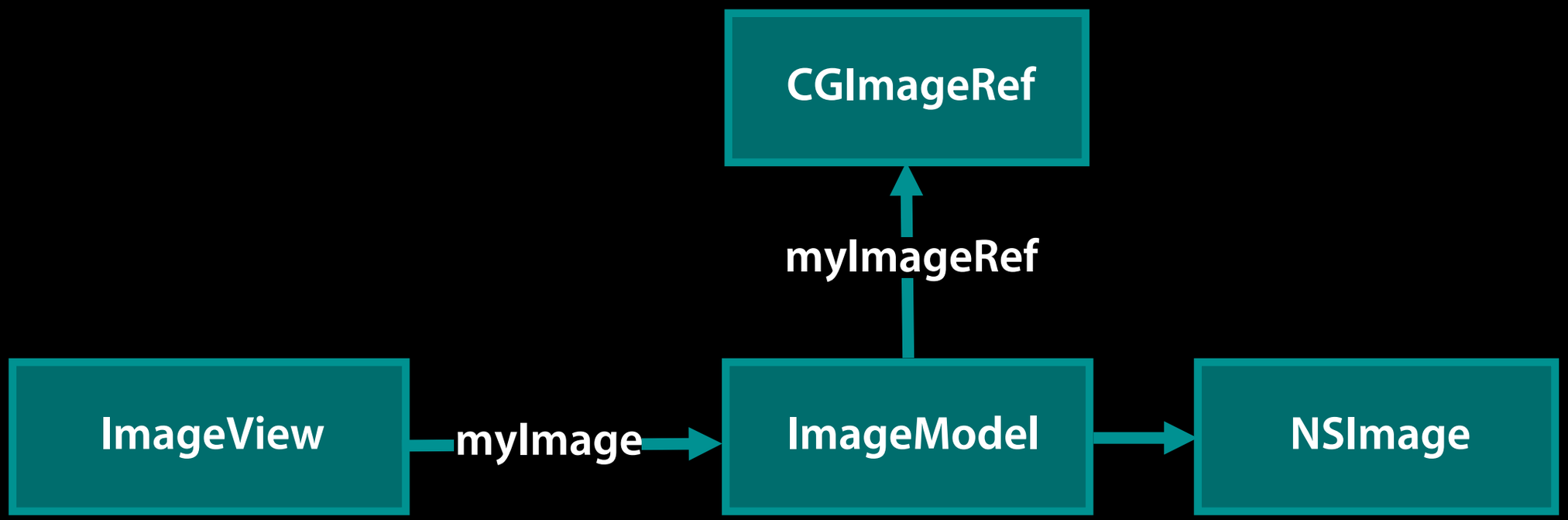


Quartz

Efficient PDF

- Optimization—write image once
- PDF “XObject” reference
- CGImageRef
  - Create via CGDataProviderCreateWithData
  - Render to CGContext with CGContextDrawImage

Quartz
Efficient PDF



Quartz
Efficient PDF

- Creating CGImageRef

```
provider = CGDataProviderCreateWithData(NULL,
  [imageRep bitmapData],
  [imageRep bytesPerRow] * size.height, NULL);

cgImage = CGImageCreate(size.width, size.height,
  [imageRep bitsPerPixel] / [imageRep samplesPerPixel],
  [imageRep bitsPerPixel], [imageRep bytesPerRow],
  CGColorSpaceCreateDeviceRGB() , kCGImageAlphaNone,
  provider, &decode, 0, kCGRenderingIntentDefault);

// drawing
CGContextDrawImage(ctx, rect, cgImage);
```

Quartz

Summary

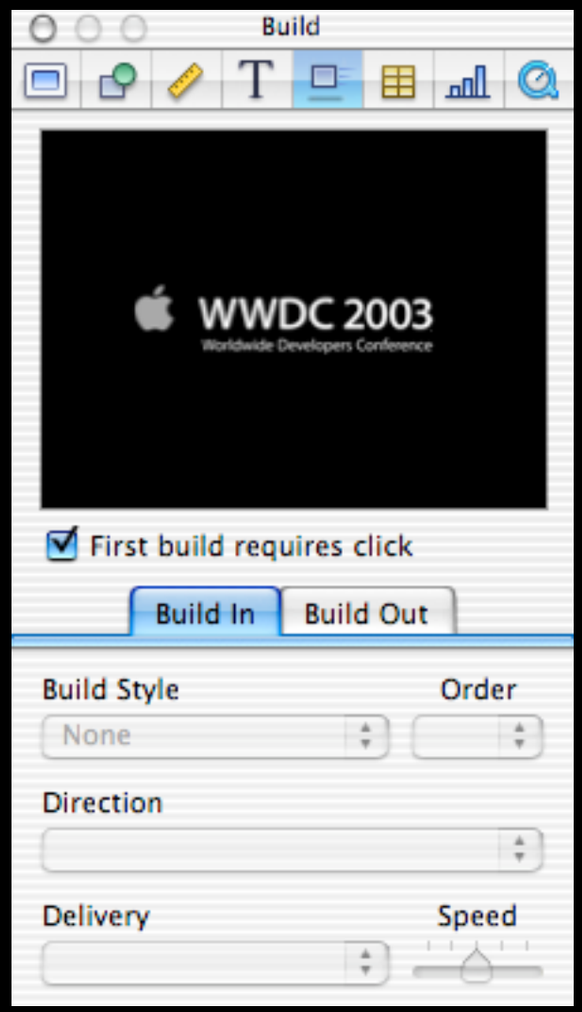
- Rich 2D capabilities
- Layered APIs
- Phenomenal performance

Coming Attractions...

Coming Attractions...

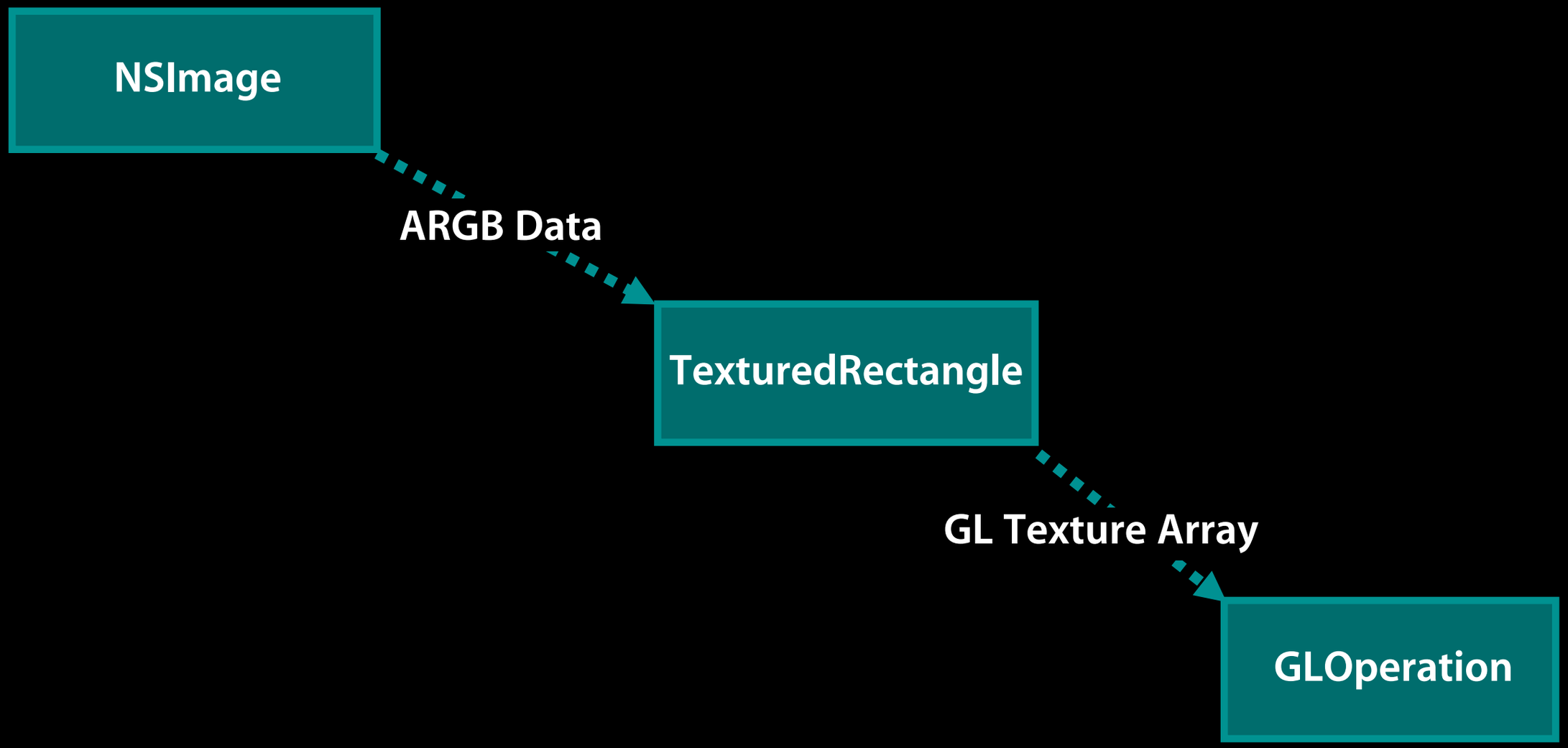
OpenGL

NSOpenGLView



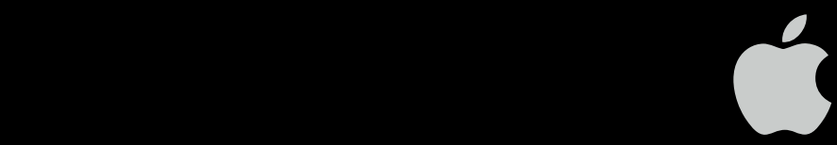
OpenGL

Quartz integration



OpenGL
Textures

```
@interface TexturedRectangle {
  texture_array_t    _textureArray;
  // ...
  GLenum             _glTarget;
}
-(id) initWithBitmap: (NSBitmapImageRep *) bitmap;
- (void)drawCorners:(NSPoint *)corners
         opacity:(float)opacity;

@end
```



OpenGL

Generating textured rectangles

- Converting NSBitmapImageRep data to ARGB

```
src = [bitmapImageRep bitmapData];

void convertRGBAtoARGB(uint *dst, uint *src, long n)
{
    while(n-->0)
    {
        uint  v;
        v       = *src++;
        v       = (v >> 8) | (v << 24);
        *dst++ = v;
    }
}
```

OpenGL

Drawing textures

```
-(void) drawCorners: (NSPoint*) corners
            opacity: (float) opacity
{
   NSPoint ul=corners[0], ur=corners[1], ll=corners[2], lr=corners[3];
   int loop_w, loop_h;

   glEnable(_glTarget);
   // Opacity support
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
   glEnable(GL_BLEND);
   glBlendFunc(GL_ONE, GL_ONE_MINUS_SRC_ALPHA);
   glColor4f(opacity, opacity, opacity, opacity);

   // Continued...
```

OpenGL

Drawing textures

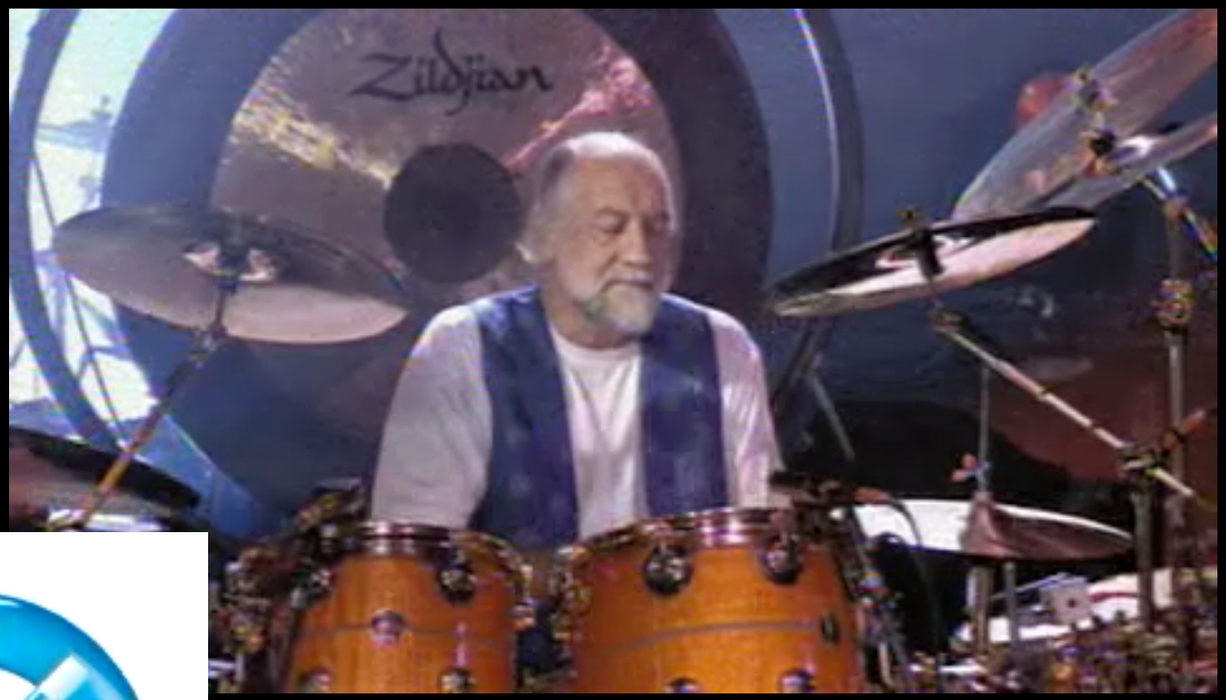
```
    for (loop_w = 0; loop_w < _textureArray.width; loop_w++)
    {
        for (loop_h = 0; loop_h < _textureArray.height; loop_h++)
        {
            texture_rect_t component = _textureArray.textures[loop_w][loop_h];
            NSPoint  iul = // interpolate points for this component...
            glBindTexture(_glTarget, component.texture_name);
            glBegin(GL_QUADS);
                glTexCoord2f(0, 0);       // upper left
                glVertex2f(iul.x, iul.y);
                glTexCoord2f(0, 1);       // lower left
                glVertex2f(ill.x, ill.y);
                glTexCoord2f(1, 1);       // lower right
                glVertex2f(ilr.x, ilr.y);
                glTexCoord2f(0, 1);       // upper right
                glVertex2f(iur.x, iur.y);
            glEnd();
        }
    }
    glDisable(GL_BLEND);

    glDisable(_glTarget);
}
```

OpenGL

Summary

- Use the GPU!
  - Quartz 2D acceleration
  - 3D effects
- NSOpenGLView
- APIs: GLUT, CGL, GL

QuickTime


Zildjian

QuickTime

QuickTime

Advantages

```
// NSImage returns filterable image formats...
// file extensions and HFS file types
+ (NSArray *)imageFileTypes
```

QuickTime

Advantages

- Image format compatibility—Free in Cocoa via NSImage
- Cocoa's NSMovieView
  - Displays a movie in a frame
- Flexible core API
  - Controlling media playback
  - Grabbing media frames

QuickTime

Play, Fast-Forward, Control Volume

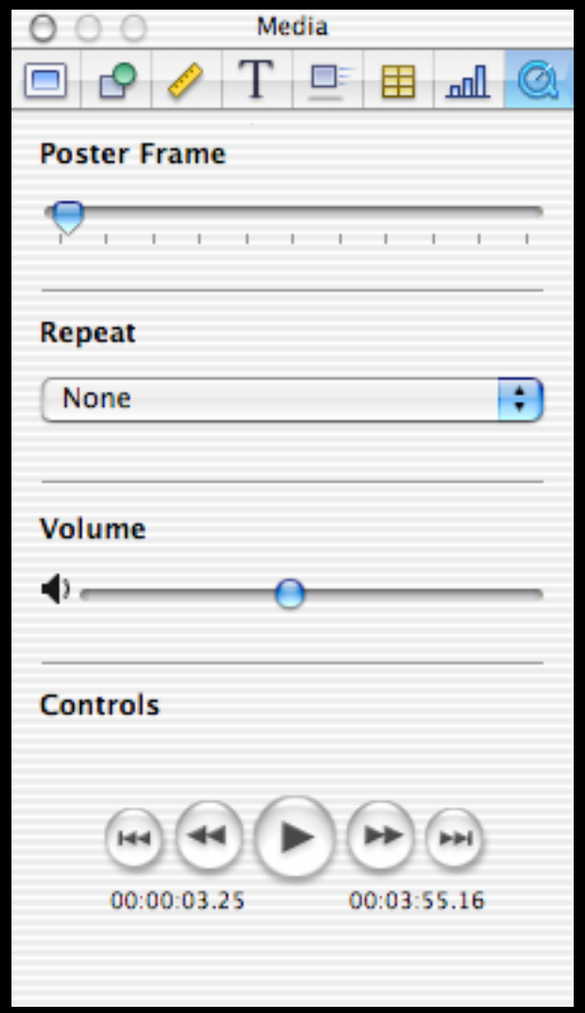
- All provided by NSMovieView

```
- (void)start:(id)sender;
- (void)setRate:(float)rate;
- (void)setVolume:(float)volume;
- (void)setMuted:(BOOL)mute;
```

QuickTime

Keynote challenge

- User control of poster frame



QuickTime
Controlling poster frame

```
// Slider action
- (IBAction) takePosterFrameFrom:(id)sender {
  // Slider value is percentage
  TimeValue  timeOffset = [sender floatValue] *
    GetMovieDuration([myMovie qtMovie]);

  // Store the poster time offset in our movie subclass
  [myMovie setPosterTime:timeOffset];
  // Uses SetMoviePosterTime(Movie, TimeValue);

  // Display that poster frame
  [myMovie setTime:timeOffset];
}
```

QuickTime

Summary

- Unrivaled compatibility
- Cocoa integration
- Rich APIs

XML

Why we love it

- Open files
- Encourage innovation
- Great for development
  - Generate suite of test documents
  - XSD/DTD validation of application output

Core Foundation XML Services

Levels of support

- Basic XML archiving with NSCoder
  - encodePropertyList: / decodePropertyList:

    `NSPropertyListFormat kCFPropertyListXMLFormat`
- High-level parser
  - Parses an entire file & returns a CFXMLTree
- Low-level parser
  - Uses callback-based API to drive XML parsing with events

Core Foundation XML Services

Keynote solution

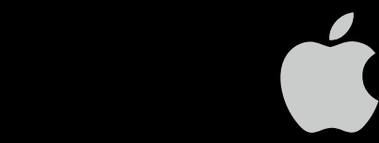
- Use low-level parser
- Generate XML with NSMutableString

```
@interface MyDOMWriter
-(void)startElement:(NSString*)in_tagName;
-(void)endElement:(NSString*)in_tagName;
-(void)writeAttribute:(NSString*)in_attributeName
value:(NSString*)in_attributeValue;
-(void)increaseIndent;
-(void)decreaseIndent;
-(void)writeDomSubtree:(MyXMLElement*)in_subtree;
// etc...
@end
```

Keynote XML

Open File Format

- Schema available at: TN2067
- Third party support
  - Filemaker
  - 4D
- Possible applications
  - Data-driven presentation generation
  - Legacy data migration (charts, tables, etc.)



4D Presentation Builder Example

Dynamic XML generation

- UI-driven database query
- Programatically creates XML
- Multimedia presentation with transitions

Demo

XML

Summary

- Great for community
- Great for development
- Great support in Mac OS X

Technology Magic of Keynote

Summary

- Cocoa application framework
- Cocoa text
- Quartz
- OpenGL
- QuickTime
- XML

Technology Magic of Keynote

Summary

- Modern technologies
- Flexible APIs
- First-rate tools
- Talented people
- Great applications!

Wrap Up

Roadmap

200 Graphics & Imaging Overview
Latest developments in Mac OS X graphics APIs.
Presidio
Tue, 10:30am

201 Mac OS X OpenGL in Depth
Introduction to OpenGL on Mac OS X.
Nob Hill
Tue, 2:00pm

202 Technology Magic of Keynote
How Keynote uses Mac OS X graphics technologies.
Nob Hill
Tue, 3:30pm

204 Image Capture Update
Integrating Image Capture with your product.
Marina
Wed, 9:00am

Roadmap

205 Vertex Programming with OpenGL Super-fast geometry transformations using the GPU.	**Marina** **Wed., 3:30pm**
206 ColorSync Update Latest developments in Mac OS X color management.	**Mission** **Wed., 5:00pm**
207 Quartz 2D in Depth Harnessing the power of Quartz 2D in your product.	**Presidio** **Thu., 9:00am**
208 Fragment Programming with OpenGL Creating incredible per-pixel effects on Mac OS X.	**Pacific Heights** **Thu., 10:30am**

Roadmap

209 OpenGL Optimizations Obtaining maximum 3D performance on Mac OS X.	**Marina** **Thu., 2:00pm**
210 Mac OS X Printing Update Latest developments in the Mac OS X printing system.	**Mission** **Thu., 3:30pm**
211 Introduction to Quartz Services Capturing & configuring displays on Mac OS X.	**Pacific Heights** **Fri., 9:00am**
212 Cutting Edge OpenGL Techniques Expert recipes for stunning 3D effects.	**Russian Hill** **Fri., 2:00pm**

Roadmap

421 Mac OS X Accessibility
Latest developments in Mac OS X accessibility.

Presidio
Fri., 10:30am

FF018 Graphics & Imaging Feedback
Share your thoughts on graphics & imaging technologies.

North Beach
Fri., 5:00pm

WWDC 2003

Worldwide Developers Conference